Discrete-time Signals and Signal Processing

A World of Signals and Signal Processing

Technological innovations have revolutionized the way we view and interact with the world around us. Editing a photo, re-mixing a song, automatically measuring and adjusting chemical concentrations in a tank: each of these tasks requires real-world data to be captured by a computer and then manipulated digitally to extract the salient information. Ever wonder how signals from the physical world are sampled, stored, and processed without losing the information required to make predictions and extract meaning from the data? **Signal processing** is the study of signals and systems that extract information from the world around us.

Signals, Defined

Perhaps the place to start the study of signal processing is the dictionary. The dictionary definition of a signal will serve us quite well: " A **signal** is "is a detectable physical quantity...by which messages or information can be transmitted."" And this information aspect is very, very critical to us. In other words, signals carry information. Signals are all around us; we encounter them throughout our day. Speech signals, for example, transmit language from one person to another via acoustic waves. If you're interested in looking for, for example, airplanes our other targets and sensing them by electromagnetic waves, you can use radar signal processing. Electrophysiological signals carry information about processes that are going on inside our bodies, things like EKGs or MRI images. And finally, financial signals transmit information about events in the economy, signals like stock prices over time or other economic markers. So as you can see, signals are really very common in the world. And this course is about signals and the signal processing systems that manipulate signals in order to understand or make transformations on the information in those signals.

Signals are Functions

Signal

> A **signal** is a function that maps an independent variable to a dependent variable.

Mathematically, we're going to think of signals as functions, and a function is just a mapping from an independent variable that we can change to a dependent variable that depends on that independent variable. The terminology is how we're going to denote a signal. It consists of an independent variable, , that for each different value of , it produces the value .

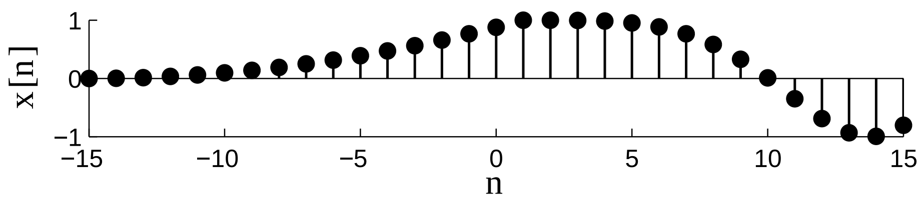
Discrete-Time Signals

Perhaps you are wondering about the use of brackets--instead of parentheses--in our signal function notation . This is the typical way to refer to **discrete-time** signals. A discrete-time signal is a signal where the independent variable is an integer (as opposed to a continuous-time signal , whose independent variable is a real number).

Plotting Discrete-Time Signals

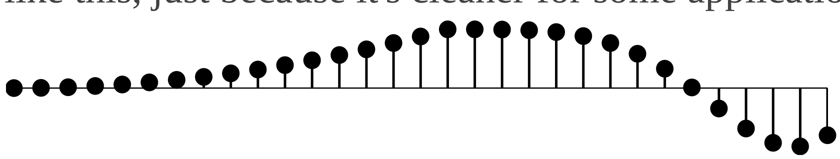
Recall that a **discrete-time signal** is a **function** with an integer-valued independent variable n. The variable n marches through time from negative infinity to positive infinity. For each value of n, we get the value of from our function $x[n]$. Now, that $x[n]$ is either going to be a real number, meaning it's going to live in the real number set, or it's going to be a complex number and live in the complex number set.

Plotting Real Signals
We're going to see a lot graphs like these in our study of signal processing:



Example of a discrete-time signal.

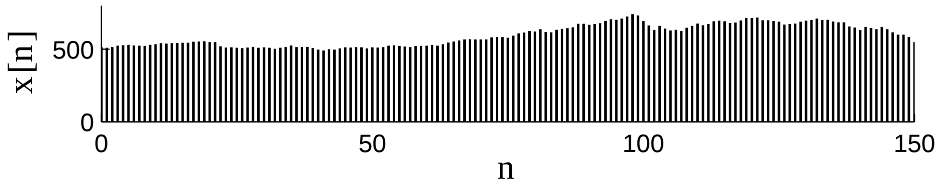
For each value of one of these n, we get the value of $x[n]$. For clarity, we're often going to color in these circles at the top, but that's really just a matter of taste. We're either going to label the signal on the y-axis or in the title of the graph. When it's clear from context that we're dealing with a discrete index n, we can strip away all of the labels and axes and just plot signals like this, just because it's cleaner for some applications:



A discrete-time signal, without the axes labeled.

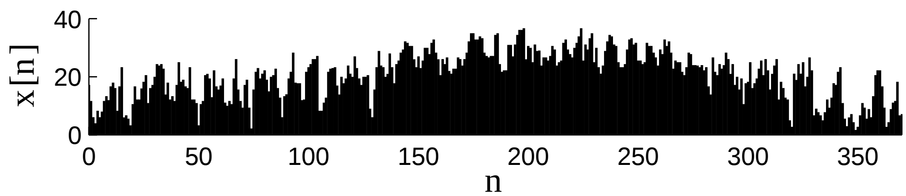
Examples of Discrete-Time Signal Plots

Here are some examples of signals. The first is a financial time series. It's the daily closing share price of Google for a five-month period:



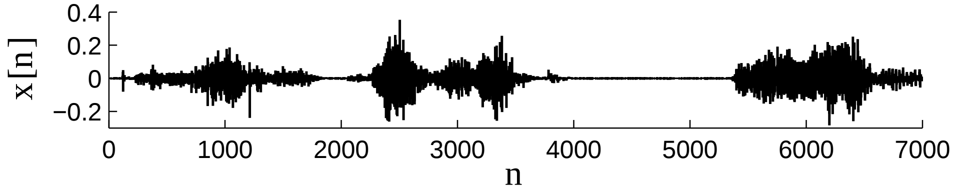
A financial series signal.

You can see here that it's a discrete time signal, where each of these signal points corresponds to one single share price at the end of a day. There are some fluctuations in the price, and if you were a financial trader or if you were an economist, you would be very interested in the information that this daily share price closing signal conveys. Another example is a temperature signal,the temperature at Houston Intercontinental Airport every day at noon for 365 days that comprise the year 2013 (in degrees Celsius):



Daily temperatures over the course of a year.

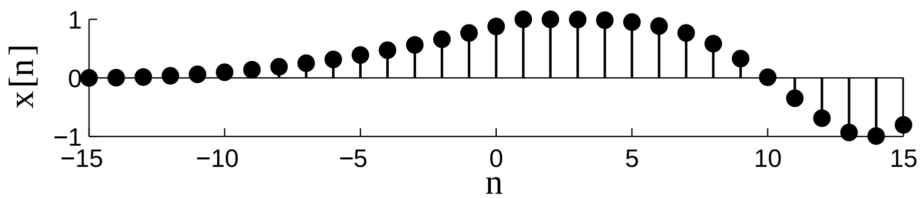
Again, we can see that there are fluctuations in this signal, and if you were a meteorologist or a climatologist, you'd be very interested in the information that this signal conveys. Finally, here's an audio signal that is speech from an actor speaking a part in Shakespeare's play, Hamlet:



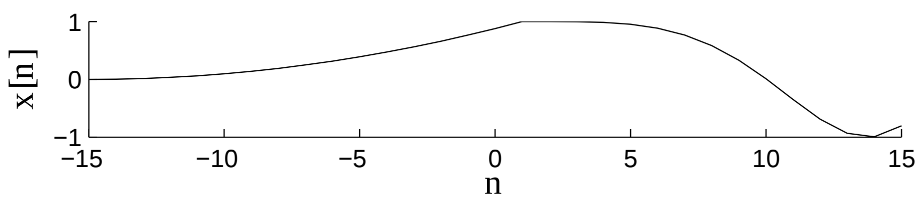
The discrete-time plot of a speech signal.

Plotting Discrete-Time Signals Correctly

We need to remember that with a discrete-time signal, the independent variable is integer valued. This means that when you plot a signal in a program like MATLAB, you must use a discrete-time plotting function (like the `stem` function) that respects the fact that the signal is only defined at discrete time points, rather than a function (like `plot`) which interpolates between points:



Discrete-time signals are undefined between the integer index values and should be plotted accordingly.



This plot interpolates between the discrete-time integer index values, which is inappropriate for a discrete-time plot.

Plotting Complex-valued Signals

Up to this point, we've been talking about real-valued signals. They comprise a single plot of n versus $x[n]$. But what about complex-valued signals?

Recall that a complex number has a real component and an imaginary component. There are two equivalent ways of expressing a given complex number. For some $a \in \quad$, we can express a in two different ways:

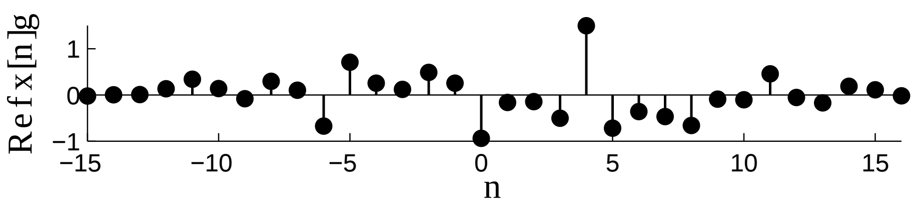
- Cartesian/rectangular form: $a = \quad (a) + j \; (a)$
- Polar form: $a = |a|e^{j\angle(a)}$,

where $j = \sqrt{-1}$ (in engineering contexts the variable j is used to represent this value because i represents electrical current). Just as a complex number can be expressed in two different ways, so can a complex-valued signal:

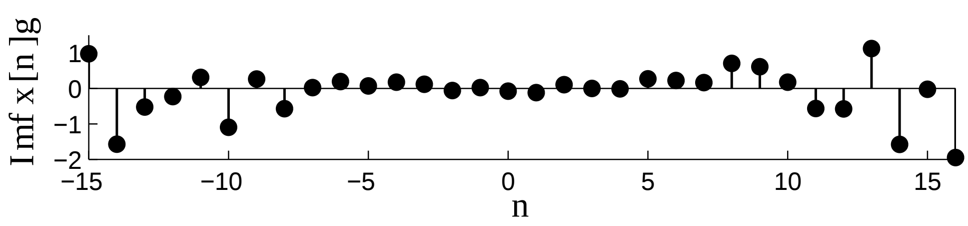
- Cartesian/rectangular form: $x[n] = \quad (x[n]) + j \; (x[n])$
- Polar form: $x[n] = |x[n]|e^{j\angle(x[n])}$

What this means is that, if we're plotting a complex-valued signal, we actually need two plots. As we have seen, there are two different ways we can plot the same complex-valued signal:

- Cartesian/rectangular form:

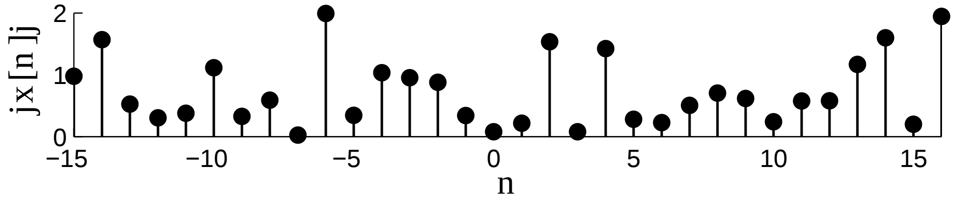


A plot of the real part of a complex-valued signal $x[n]$.

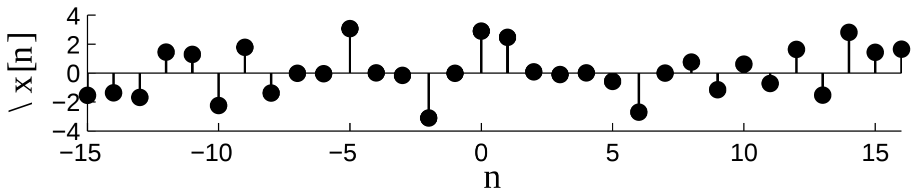


A plot of the imaginary part of a complex-valued signal $x[n]$.

- Polar form:



A plot of the magnitude of a complex-valued signal $x[n]$. Note how all of the values are greater than or equal to 0.



A plot of the phase of a complex-valued signal $x[n]$. Note how the values range between $-\pi$ and π.

Signal Properties

Signal Classification

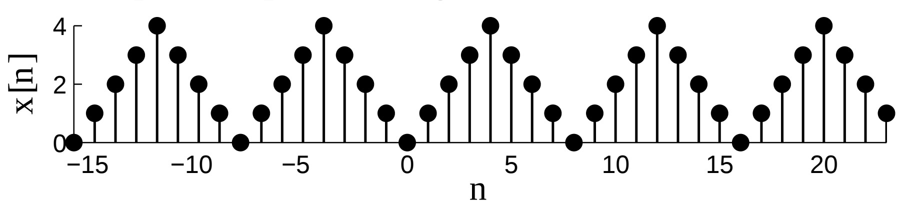
Signals can be broadly classified as discrete-time or continuous-time, depending on whether the independent variable is integer-valued or real-valued. Signals may also be either real-valued or complex-valued. We will now consider some of the other ways we can classify signals.

Signal Length: Finite/Infinite

This classification is just as it sounds. An **infinite-length** discrete-time signal takes values for all time indices: all integer values n on the number line from $-\infty$ all the way up to ∞. A **finite-length** signal is defined only for a certain range of n, from some N_1 to N_2. The signal is not defined outside of that range.

Signal Periodicity

As the name suggests, **periodic** signals are those that repeat themselves. Mathematically, this means that there exists some integer value N for which $x[n+N]=x[n]$, for all values of n. So if we define a fundamental period of this particular signal of length, like $N=8$, then we will see the same signal values shifted by 8 time indices, by 16, -8, -16, etc. Below is an example of a periodic signal:



A periodic discrete-time signal. Note how it repeats every 8 time units.

So periodic signals repeat, and clearly periodic signals are going to be, therefore, infinite in length. It's also important to remember that to be periodic in discrete-time, the period N must be an integer. If there is no such integer-valued N for which $x[n+N]=x[n]$ (for all values of n), then we classify the signal as being **aperiodic**.

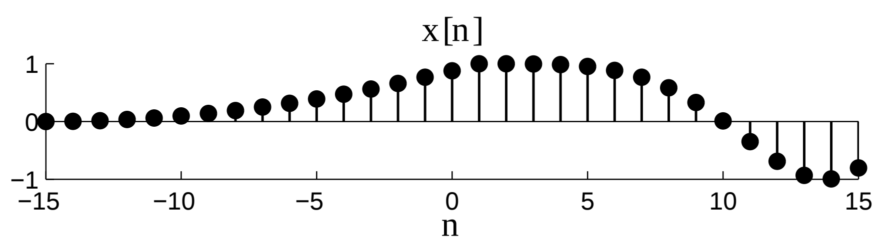
Converting Between Infinite and Finite Length
In different applications, the need will arise to convert a signal from infinite-length to finite-length, and vice versa. There are many ways this operation can be accomplished, but we will consider the most common.

The most straightforward way to create a finite-length signal from an infinite-length one is through the process of **windowing**. A windowing operation extracts a contiguous portion of an infinite-length signal, that portion becoming the new finite-length signal. Sometimes a window will also scale the smaller portion in a particular way. Below is a mathematical expression of windowing (without any scaling):

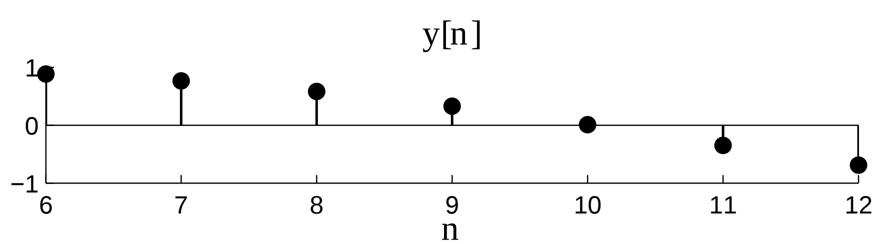
$$y[n] = \begin{cases} x[n] \ \text{ if } \ N_1 \leq n \leq N_2 \\ \text{undefined} \ \text{ if } \ \text{else} \end{cases}$$

Below is a signal $x[n]$ (assume it is infinite-length, with only a part of it shown), with a portion of it extracted to create $y[n]$:

(a) an infinite-length signal $x[n]$ (only part of it shown) has a portion extracted via windowing to create (b) a finite-length signal $y[n]$.



Infinite-length signal (only a portion of it is shown)



Finite-length signal

There are two ways a signal can be converted from a finite-length to infinite-length. The first is referred to as **zero-padding**. It is easy to take a finite-length signal and then make a larger finite-length signal out of it: just extend the time axis. We have to decide what values to put in the new time locations, and simply putting 0 at all the new locations is a common approach. Here is how it looks, mathematically, to create a longer signal $y[n]$ from a shorter signal $x[n]$ defined only on $N_1 \leq n \leq N_2$:

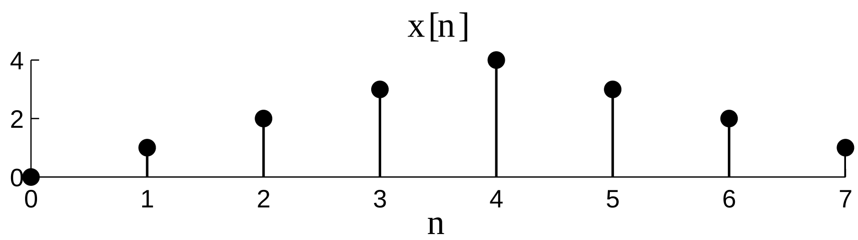
$$y[n] = \begin{array}{l} 0 \text{ if } N_0 \leq n < N_1 \\ x[n] \text{ if } N_1 \leq n \leq N_2 \\ 0 \text{ if } N_2 < n \leq N_3 \end{array}$$

Here, obviously $N_0 < N_1 < N_2 < N_3$, and if we extend N_0 and N_3 to negative and positive infinity, respectively, then $y[n]$ will end up being infinite-length.

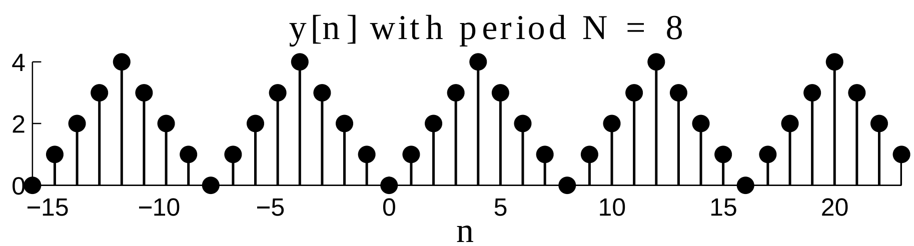
The other way to make an infinite-length signal from a finite-length one is to **periodize** it, which means replicating a finite-length signal over and over to create an infinite-length periodic version. Mathematically, that means defining the new infinite-length periodic signal like this:`\begin{eqnarray*} y[n] &=& \sum_{m=-\infty}^{\infty} x[n-mN], \quad n\in\Integers \\ &=& \cdots + x[n+2N] + x[n+N] + x[n] + x[n-N] + x[n-2N] + \cdots \end{eqnarray*}`

Graphically, we can see that this amounts to repeating the signal over and over, before and after the original portion:

(a) finite-length signal $x[n]$ is periodized to create (b) an infinite-length signal $y[n]$ (only a portion of it is shown).

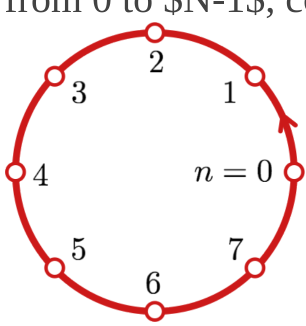


Finite-length signal



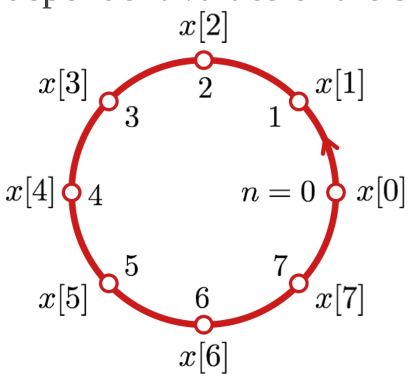
Original signal periodized to create an infinite-length signal.

Periodization and Modular Arithmetic
It turns out that, as we consider periodization and periodic signals, the notion of modular arithmetic will be helpful. In modular arithetic, integers do not lie on a line stretching from negative infinity to infinity, but rather on a circle of a defined size N. In modulo-8, for example, the numbers are 8 "hours" on a "clock." Our convention will be for the numbers to traverse from 0 to $N-1$, counterclockwise:



CAPTION.

Consider a finite-length signal of size N. We can align the time-dependent values of the signal on the modulo circle:



CAPTION.

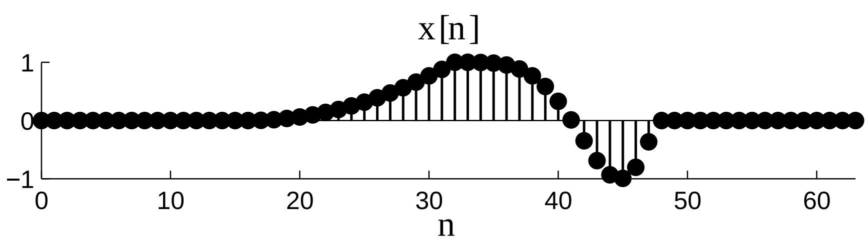
When we travel around the clock once, from time index 0 to 7, we express the finite-length signal. But if we keep traveling, in one direction or the other, then that amounts to periodizing the signal. Using our modulo notation, we can periodize a finite-length signal $x[n]$ to be an infinite-length periodic signal $y[n]$ like this: $y[n]=x[(n)_N]$.

Finite/Periodic Signals Relationship
We have seen that we can take an N-length finite-length signal and periodize it to make an infinite-length periodic signal with a period of N. By the same token, we can also work in reverse and extract one period worth of signal from any periodic signal to create a finite-length signal. **What this means is we can consider periodic signals and finite-length signals to be essentially equivalent**: we can consider just one period of a periodic signal (the rest of the signal is redundant, by definition), or periodize a finite-length signal. They are two ways of looking at the same thing, a phenomenon we will often see in our study of signals and systems, and we will choose the perspective that best suits our needs for particular applications.

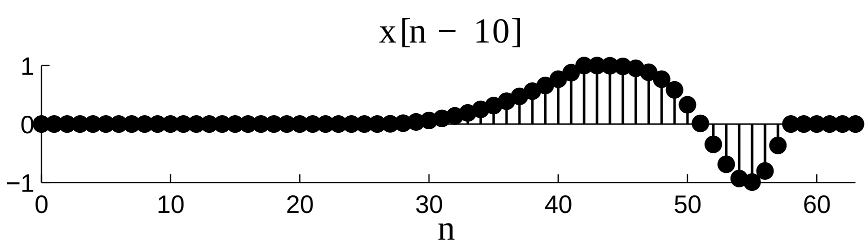
Shifting Infinite-length Signals

Given some signal $x[n]$, it will often be necessary for us to consider that signal shifted in time. We denote such a shift mathematically with an expression like $x[n-m]$, where m is some integer. If m is greater than zero, then $x[n-m]$ will be just like $x[n]$, except it will be shifted to the right by m time units. If m is less than zero, it will be shifted to the left. Here is who that might look for a couple values of m:

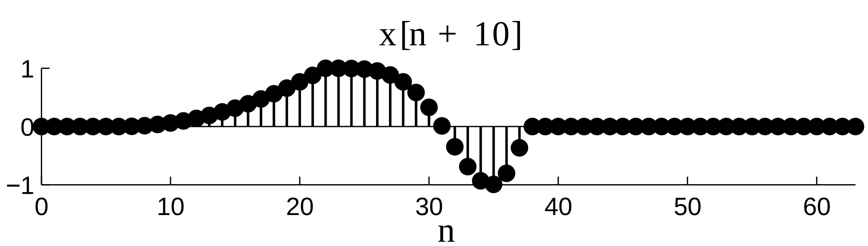
A (a) signal $x[n]$ shifted according to the expression $x[n-m]$ where m is (b) positive, and (c) negative.



Original signal $x[n]$.



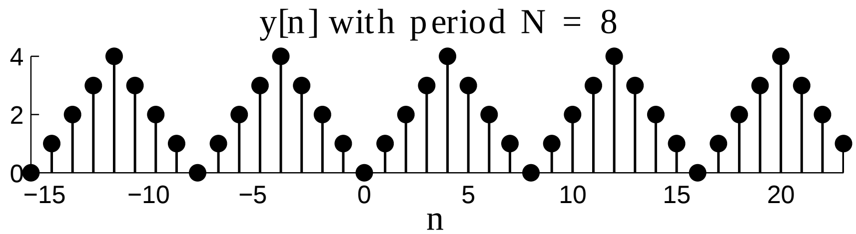
$x[n]$ shifted to the right.



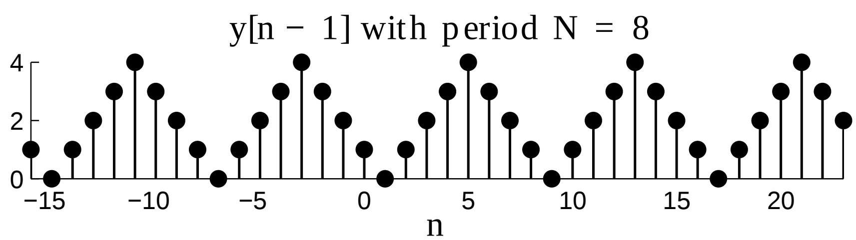
$x[n]$ shifted to the left.

This type of shifting works the same with periodic signals:

A periodic signal shifted one value in time.



Original periodic signal $y[n]$.



$y[n]$ shifted to the right.

Of course, for periodic signals, certain shifts actually do not have any effect on them. If a signal repeats with a period of N, then shifting that signal by any integer multiple of N will yield the original signal. Take a look at the signal above, which was shifted to the right by a time unit of 1. If we keep on shifting it until it is shifted 8 time units the result will be identical to the original signal.

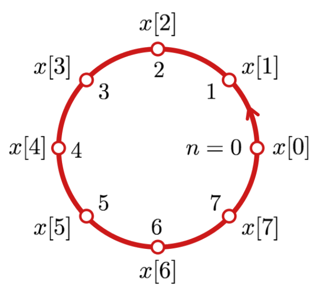
Shifting Finite-Length Signals

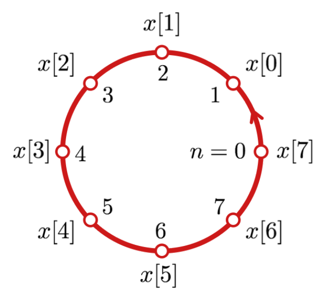
Can finite-length signals be shifted, as well? There does not seem to be any reason why not. Suppose we have some finite-length signal $x[n]$ (of length N) and we define another finite-length signal $v[n]$ to be $v[n]=x[n-1]$. So we have $v[1]=x[0]$ and $v[2]=x[1]$ and so on until

$v[N-1]=x[N-2]$. But what about $v[0]$, what do we put there? And how about $x[N-1]$, where is that supposed to go? We do not want to invent information to put in $v[0]$, nor lose the information of $x[N-1]$. An elegant solution is to periodize $x[n]$, and then consider the relation $v[n]=x[n-1]$. In this case, we now have a value for $v[0]$: $v[0]=x[-1]$. Since $x[n]$ is periodic with period N, it also happens that $x[-1]=x[N-1]$, so we do not lose that information.

This kind of operation, for finite-length signals, is called a **circular shift**, and we can express it mathematically with the help of our modular arithmetic operator. Circularly shifting a finite-length signal $x[n]$ by m time units is expressed as $x[(n-m)_N]$. It can also be visualized by turning $x[n]$ about the circle on which it resides:

Original finite-length signal $x[n]$. $x[(n-3)_8]$

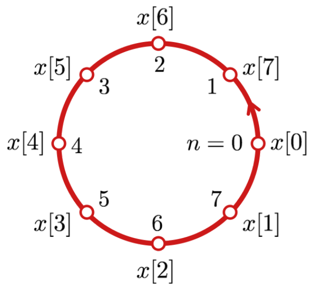




Circularly shifting a signal by m amounts to turning it counter-clockwise m steps.

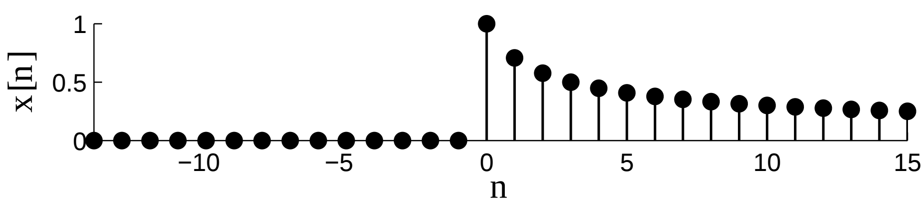
Time Reversing Finite-Length Signals

For infinite length signals, the transformation of reversing the time axis $x[−n]$ is obvious: just flip the signal about $n=0$. But things are not quite so obvious for finite-length signals; if a signal is defined for, say, n between 0 and N, then what gets flipped across the $n=0$ from the negative side? Once again, it turns out the modular arithmetic opererator can be called in for help. We reverse the time axis, modulo N: $x[(-n)_N]$. Below is an image of a finite-length ($N=8$) signal, time-reversed:

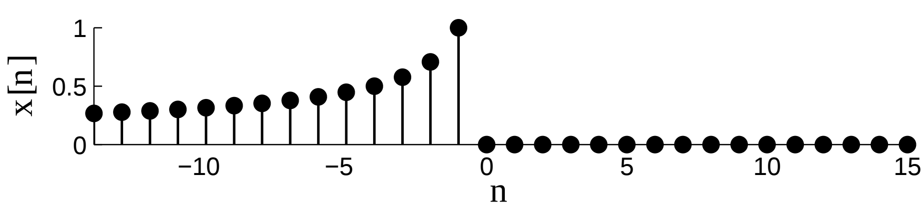


We can time reverse a finite-length signal $x[n]$ with the mathematical expression $x[(-n)]_8$.

Signal Causality
A signal $x[n]$ is **causal** if $x[n]=0$ for all $n \lt 0$, it is **anti-causal** if $x[n]=0$ for all $n \geq 0$, and it is **acausal** if is neither causal nor anti-causal.

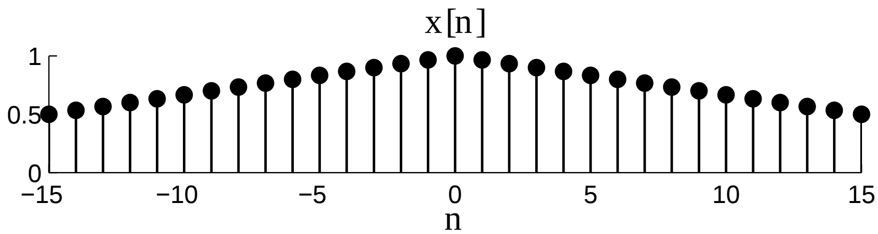


A causal signal.

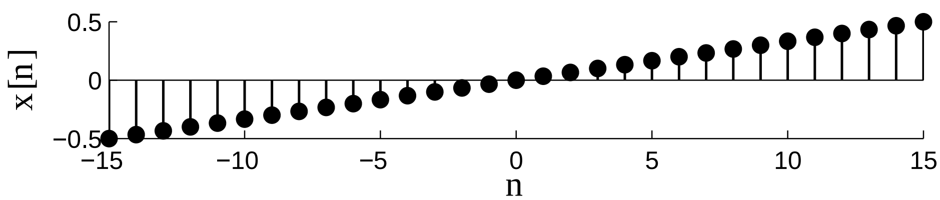


An anti-causal signal.

Even and Odd Signals
A signal $x[n]$is defined as **even** if $x[-n]=x[n]$, and **odd** if $x[-n]=-x[n]$.



An even signal.

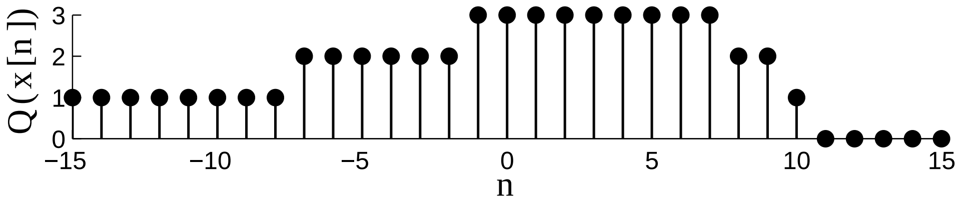


An odd signal.

Even/Odd Signal Decomposition
MM: Is it really necessary to include this? Usage of the concept (wavelets?) is outside the scope of the class.

Digital Signals
Digital signals are a special sub-class of discrete-time signals. While the independent time variable for discrete-time signals is integer-valued, the dependent variable (i.e., the value the signal takes at any given time) can take on any value. However, for digital signals, both of these variables are discrete-valued. Rather than take any value on a continuum, discrete signals take only a limited number of values, or **levels**. Typically, the number of

levels is expressed as $D = 2^q$, and each possible value of $x[n]$ is represented as a digital code with q bits.

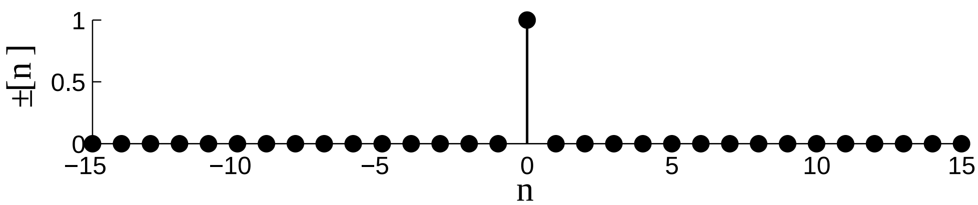


A digital signal with $q = 2$ bits, so $D = 2^2= 4$ levels.

Key Discrete-time Test Signals

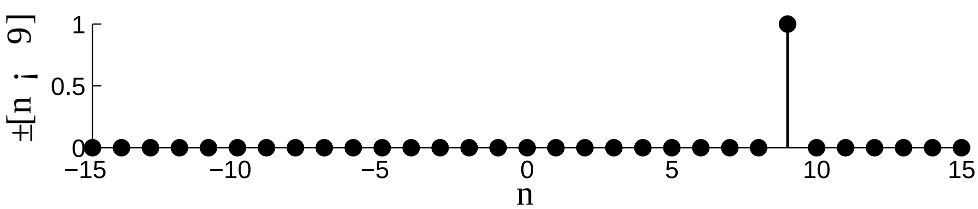
In our study of discrete-time signals and signal processing, there are five very important signals that we will use to both illustrate signal processing concepts, and also to probe or test signal processing systems: the **delta function**, the **unit step function**, the **unit pulse function**, the **real exponential function**, **sinusoidal functions**, and **complex exponential functions**. This module will consider the first four; sinusoids and complex exponentials are particularly important, so a separate model will cover them. Each of these signals will be introduced as infinite-length signals, but they all have straightforward finite-length equivalents.

The Discrete-time Delta Function
The delta function is probably the simplest nontrivial signal. It is represented mathematically with (no surprise) the Greek letter delta: $\delta[n]$. It takes the value 0 for all time points, except at the time point 0 where it peaks up to the value 1: $\delta[n]=\begin{cases}1 & n=0 \\ 0 & \textrm{otherwise}\end{cases}$



The discrete-time delta function.

In a variety of important settings, we will often see the delta function shifted by a particular time value. The delta function $\delta[n-m]$ is 0, except for a peak of 1 at time m:

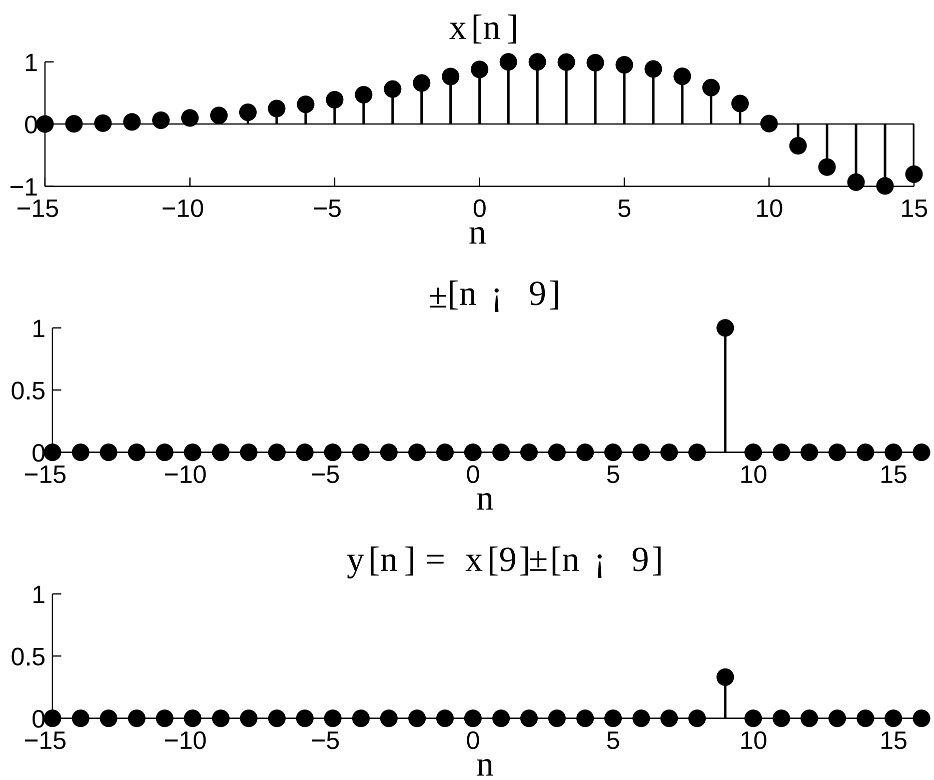


A time-shifted discrete-time delta function $\delta[n-

m]$, where $m=9$.

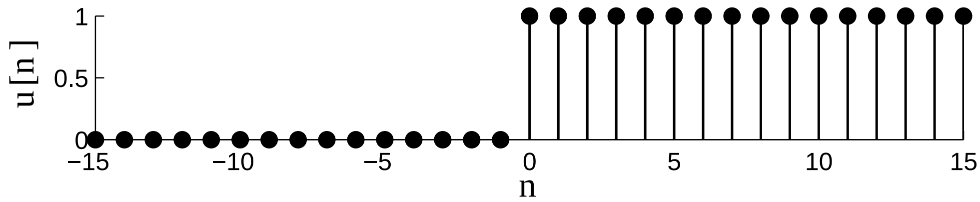
One of the reasons the shifted delta function is so useful is that we can use it to select, or sample, a value of another signal at some defined time value. Suppose we have some signal $x[n]$, and we would like to isolate that signal's value at time m. What we can do is multiply that signal by a shifted delta signal. We can say $y[n]=x[n]\delta[n-m]$, but since that $y[n]$ will be zero for all n except at $n=m$, it is equivalent to express it as $y[n]=x[m]\delta[n-m]$, where now $x[m]$ is no longer a function, but a constant. The following figure shows how this operation isolates a particular time sample of $x[n]$:

Using a time-shifted delta function to isolate a sample of the signal $x[n]$.



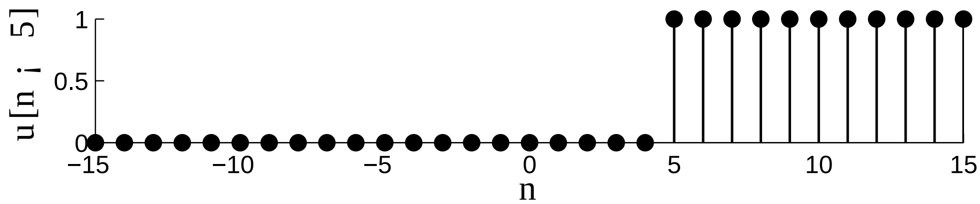
The Unit Step Function

The unit step function can be thought of like turning on a switch. Usually identified as $u[n]$, it is 0 for all $n \lt 0$, and then at $n=0$ it "switches on" and is 1 for all $n \geq 0$: $u[n]=\begin{cases}1 & n \lt 0\\ 1 & n\geq 0\end{cases}$:



The unit step function.

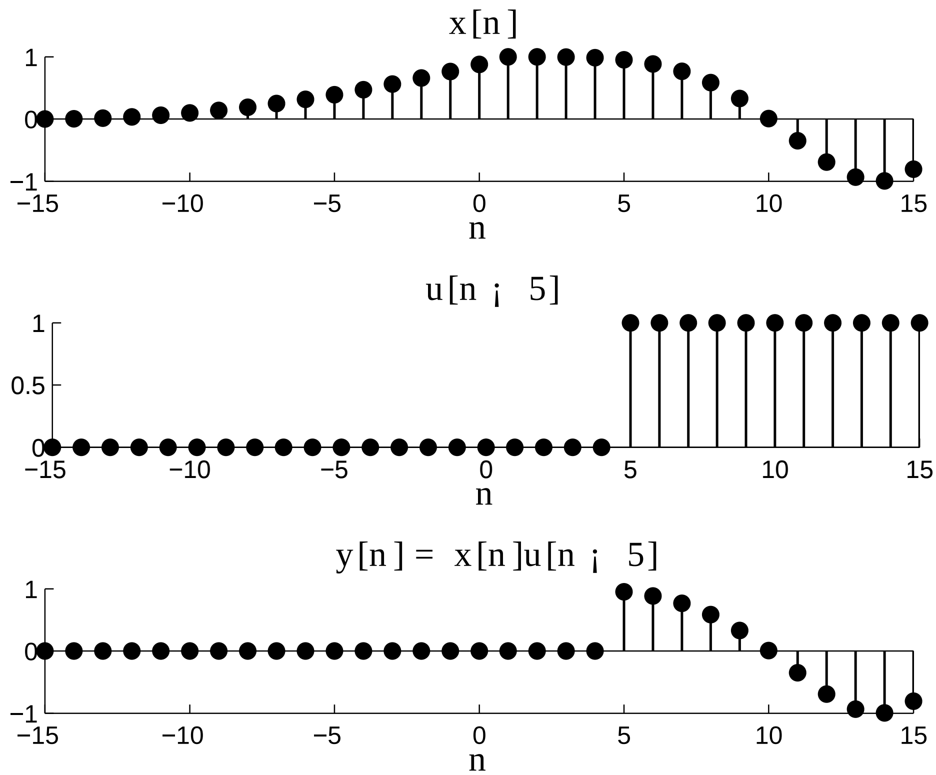
As with the delta function, it will also be useful for us to shift the step function:



A shifted step function $[n-m]$ with $m=5$.

And, as you might have guessed, we can use a shifted step function in a similar way to the delta function by multiplying it with another signal. Whereas the delta function selected a single value of a certain signal (zeroing out the rest), the step function isolates a portion of a signal after a given time. Below, a step function is used to zero out all the values of $x[n]$ for $n\lt 5$, keeping the rest:

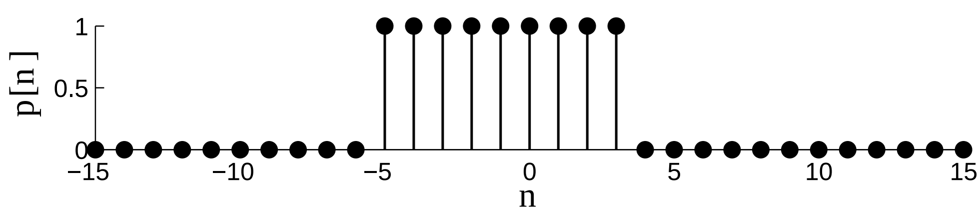
A shifted step function can be used to zero out all values of a signal before a certain time index.



Supposing a signal $x[n]$ were not causal, setting m to zero and performing the operation $x[n]u[n]$ would zero out all values of $x[n]$ before $n=0$, thereby making the result causal.

The Unit Pulse Function

The unit pulse $p[n]$ is very similar to the unit step function in how it "switches on" from 0 to 1, but then it also "switches off" at a later time. We will say it "switches on" at time N_1, and "off" at time N_2: $p[n] = \begin{cases} 0 & n\lt N_1 \\ 1 & N_1 \leq n \leq N_2 \\ 0 & n\gt N_2\\ \end{cases}$

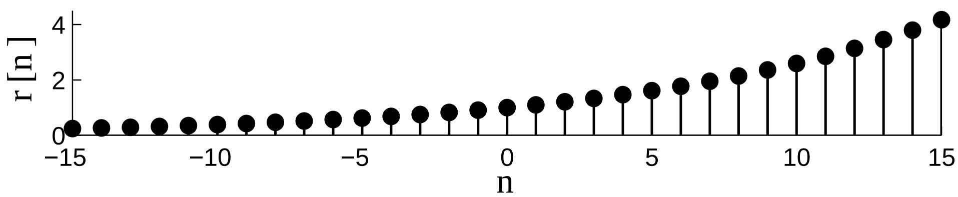


The unit pulse function $p[n]$, here with $N_1 = −5$ and $N_2 = 3$.

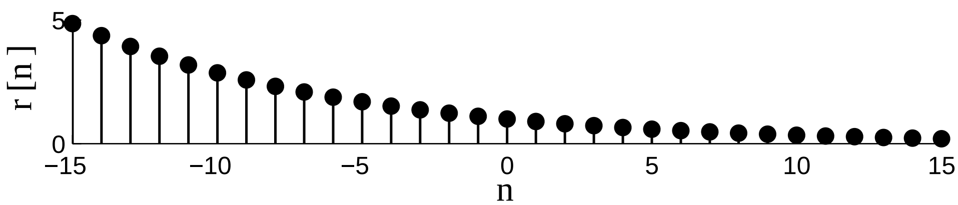
Of course, rather than use the above piece-wise notation, it is also possible to express the pulse as the difference of two step functions: $p[n] = u[n-N_1] - u[n-(N_2+1)]$.

The Real Exponential Function

Finally, we have the real exponential function, which takes a real number a (that we are going to assume is positive) and raises it to the power of $n,$ where n is the time index: $r[n] = a^n$, $a\in R$, $a\geq 0$. So at $n=0$, $r[n]=a^0$, at $n=1$ it equals a, is a^2 at $n=2$, and so on. As the name suggests, the signal will exponentially increase or decrease, depending on the value of a.



For $a\gt 1$, the real exponential function increases with time. Here $a=1.1$.



For $0 \lt a\lt 1$, the real exponential function decreases with time (or we could say it increases exponentially as the time index decreases). Here $a=.9$.

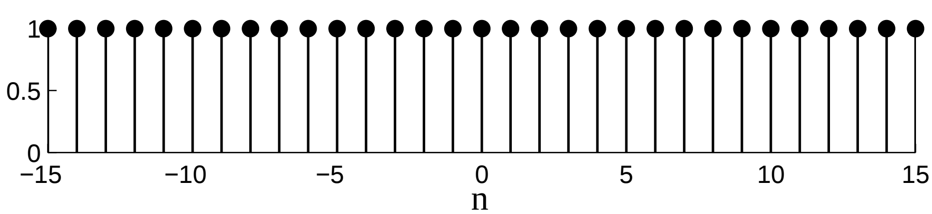
Real and Complex Sinusoidal Signals

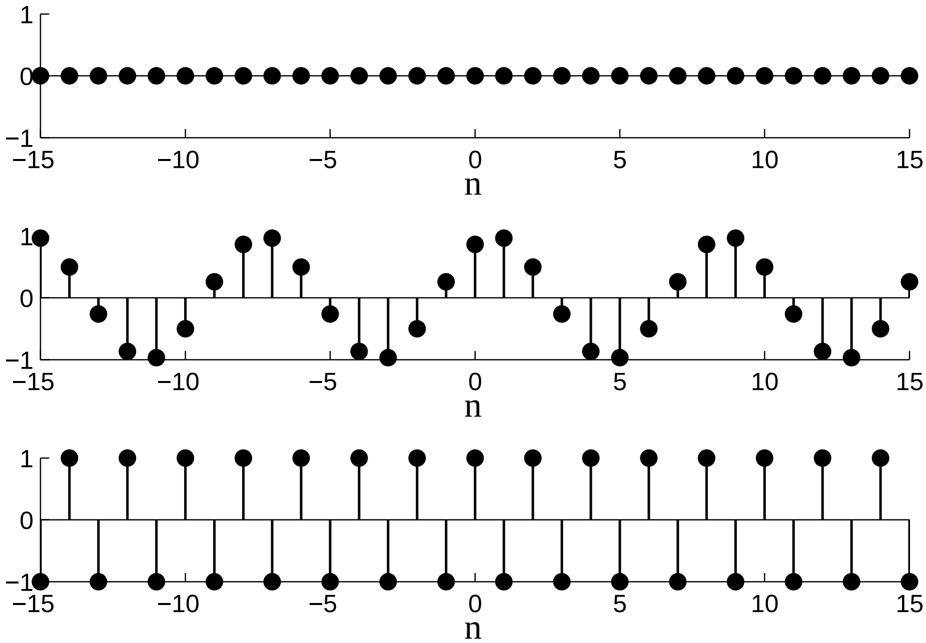
Discrete-time real and complex valued sinusoidal signals are an incredibly important signal class in the study of discrete-time signals and systems. Of course, sinusoidal waves show up in all sorts of science and engineering applications, but they are particularly relevant for signal processing because they are the foundation of Fourier analysis.

Real Valued Sinusoids

There are two real-valued discrete-time sinusoidal wave signals: the **sine** wave signal and the **cosine** wave signal. They are represented mathematically as $\sin(\omega n +\phi)$ and $\cos(\omega n +\phi)$. Let's take a look a those in more detail. First, as we have seen with other discrete-time signals, n is the independent variable time index, and it runs from negative infinity to infinity. The variable ω is known as the **frequency** of the sinusoidal signal, and we will see how changing the value of ω impacts the rate of the signal's oscillation. The variable ϕ is the **phase** of the signal, and changing it will shift the signal left along the time axis. Finally, the terms $\sin$ or $\cos$ return the corresponding trigonometric value to $\omega n +\phi$ for each value of the time index n. Here are a few examples of real sinusoidal waves:

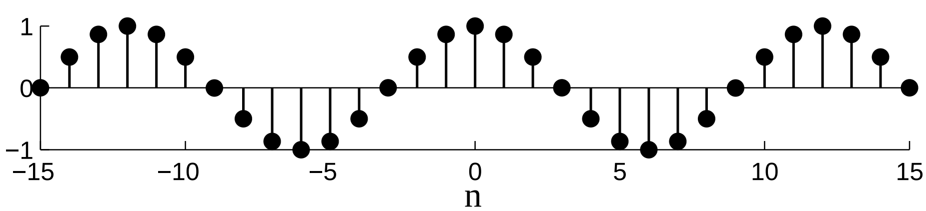
(a) A plot of $\cos(0n)$. At every point in time, this signal takes the value $\cos(0)=1$. (b) A plot of $\sin(0n)$. At every point in time, this signal takes the value $\sin(0)=0$. (c) A plot of $\sin(\frac{\pi}{4}n+\frac{2\pi}{6})$. (d) A plot of $\cos(\pi n)$. Note how when $\omega=\pi$, as in this example, the signal is oscillating as rapidly as possible, between -1 and 1 at every single time instance. This phenomenon is the opposite of when $\omega=0$, for which the signal does not oscillate at all. So in some sense we can see that 0 is the lowest possible frequency ω, and π is the highest.



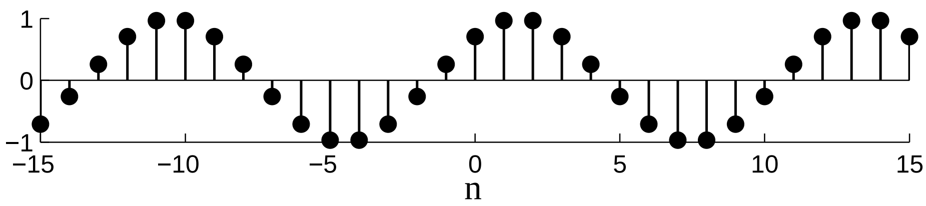


We saw in the figure above how the frequency ω influences the rate of the wave's oscillation. The other variable in the signal, the phase ϕ, can shift the wave backwards and forwards along the time axis, without affecting the frequency. Below are plots of a cosine wave which all have the same frequency, but with a variety of phase shifts (i.e., different values of ϕ):

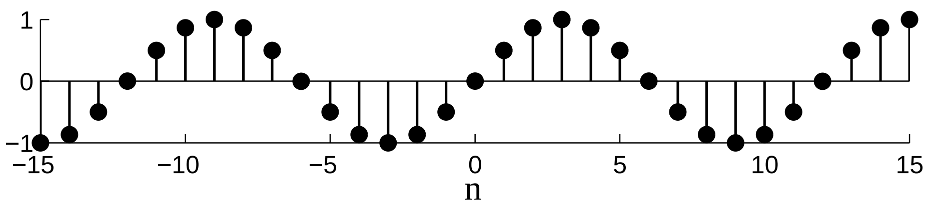
Four cosine waves with the same frequency, but different phases. Note how a phase shift of $\frac{\pi}{2}$ shifts the cosine to be a sine wave, and a phase shift of 2π shifts it all the way over to where it was without any phase shift (because the cosine is periodic for this frequency).



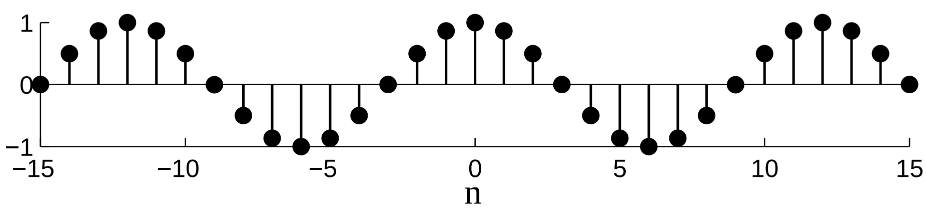
$\cos\left(\frac{\pi}{6}n-0\right)$.



$\cos\left(\frac{\pi}{6}n-\frac{\pi}{4}\right)$.



$\cos\left(\frac{\pi}{6}n-\frac{\pi}{2}\right) = \sin\left(\frac{\pi}{6}n\right)$.

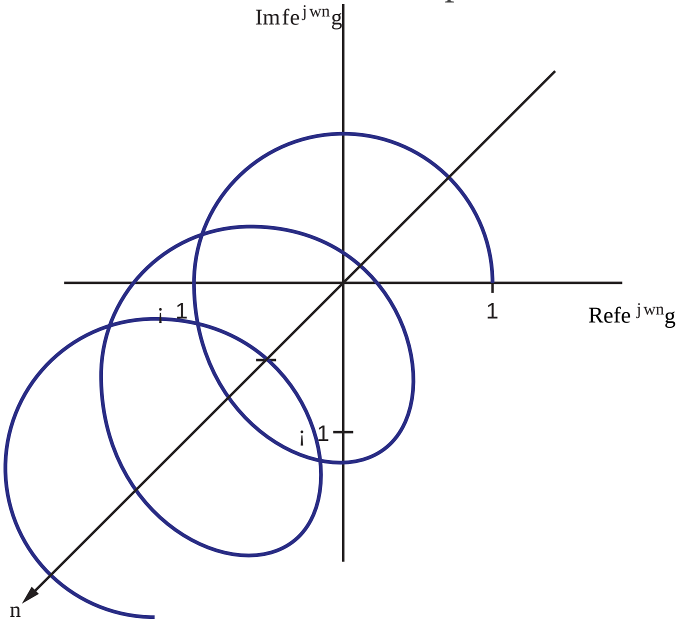


$\cos\left(\frac{\pi}{6}n-2\pi\right) = \cos\left(\frac{\pi}{6}n\right)$.

Complex Valued Sinusoids

So we have reviewed the real sine waves $\sin$ and $\cos$, and perhaps seeing them in proximity brought to mind a very special relationship called **Euler's Formula**: $e^{j\theta}=\cos(\theta)+j\sin(\theta)$ (you may remember this from math class with an i instead, but recall engineers use that letter for current, and we call the imaginary number j). That formula works for any particular value of θ, so of course it applies when we consider $\omega n+\phi$, as above, which gives us a complex valued

sinusoid: $e^{j(\omega n +\phi)}=\cos(\omega n +\phi)+j\sin(\omega n +\phi)$. Let's look at some plots of complex sinusoids. Unlike two-dimensional real sinusoids (which have an one-dimensional independent time variable n and a take a one-dimensional value at each time value), complex sinusoids are three dimensional: they have the time dimension, a real dimension, and an imaginary dimension. So they can be visualized as a three dimensional helix in space:



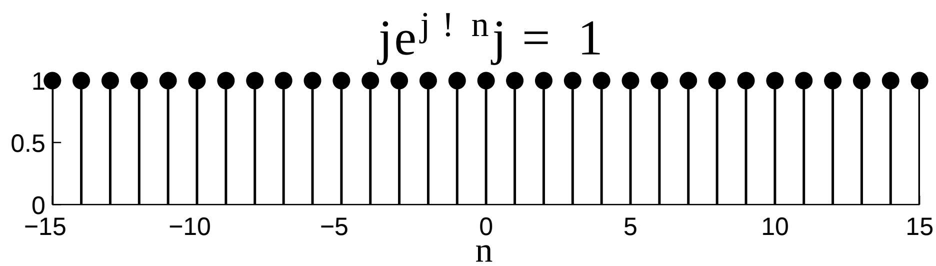
A three dimensional visualization of a complex sinusoid. Note that this image is continuous-valued, whereas a discrete valued version would actually appear as points along the line.

If you were to look at this helix from directly above, you would see only the real portion of the helix, and it would appear to be a cosine wave. If you looked at it from the side, you would see the imaginary aspect of it, as a sine wave. The frequency variable ω controls how quickly the helix rotates across time n, and also the direction: positive values cause it to

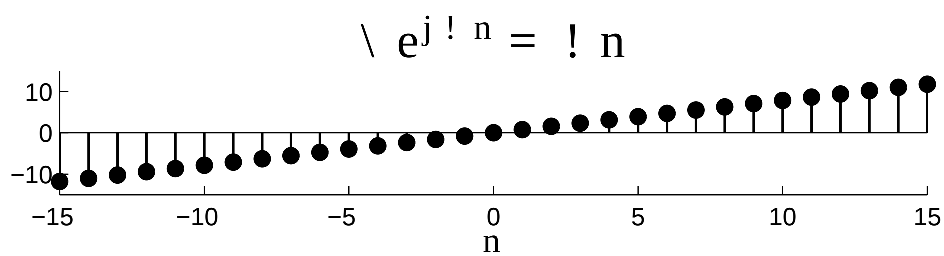
rotate in the counterclockwise manner shown, and negative values would result in it rotating clockwise.

While it is illuminating to visualize complex simusiods in three dimensions, in practice it is actually most common to view them in two, separately plotting either the real and the imaginary parts with respect to time, or the magnitude and phase across time:

A complex sinusoid plotted according to its magnitude and phase. Note that the magnitude of a single complex sinusoid is trivial, as $|e^{j(\omega n +\phi)}|=1$.

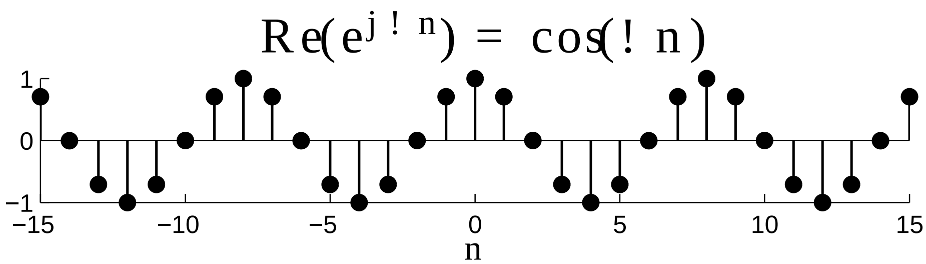


The magnitude of a complex sinusoid.

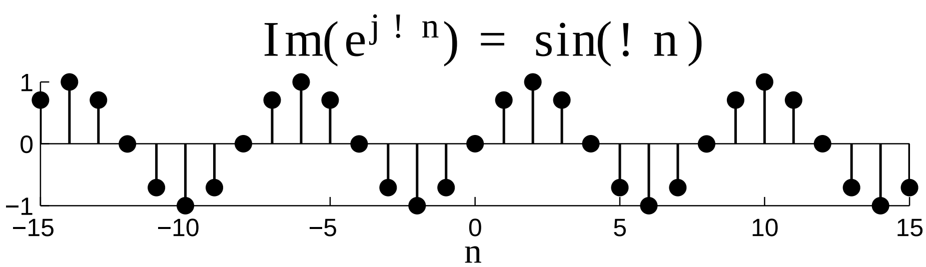


The phase, or angle, of a complex sinusoid.

A complex sinusoid plotted according to its real and imaginary parts. These are a cosine, and sine, respectively, which follows from Euler's Formula.



The real part of a complex sinusoid.



The imaginary part of a complex sinusoid.

We'll wrap up our introduction of sinusoids by briefly considering the concept of negative-valued frequencies. It is easiest to see the difference a negative frequency makes, compared to a positive frequency of the same magnitude, by expressing it all mathematically: $e^{j (-\omega) n} ~=~ e^{-j\omega n} ~=~ \cos(-\omega n) + j \sin(-\omega n) ~=~ \cos(\omega n) - j \sin(\omega n)$ So negating the frequency of a complex sinusoid has no effect on the real part of the signal (the cosine), but it flips the sign of the imaginary part (the sine). This operation (preserving the real part, but changing the sign of the imaginary part) is also known as taking the complex conjugate of the signal. So negating the frequency of a complex sinusoid is the same thing as taking the complex conjugate of it:$e^{j (-\omega) n} ~=~ e^{-j \omega n}~=~ \left(e^{j \omega n} \right)^*$.

Why use imaginary numbers?

Now perhaps you are wondering the point of using imaginary numbers. After all, aren't all real world signals, well, real-valued? They are indeed,

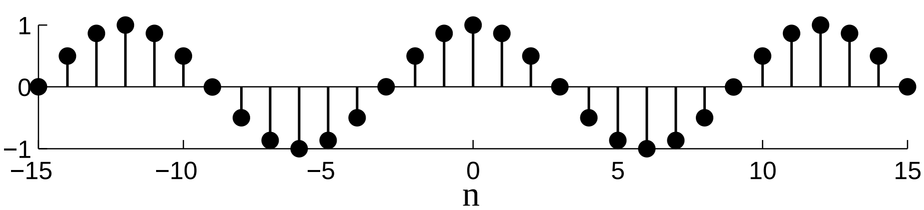
but we can consider them as the real-part of a complex-valued signal. And why go to that trouble? There are many good reasons, but here is one to start with: exponential functions are much easier to work with than trigonometric functions. You can easily simplify $e^{a} e^{b}$ into a single term, but you very likely would be turning to a table to simplify $\sin(a)\cos(b)$, wouldn't you?

The Peculiarity of Discrete-Time Sinusoids

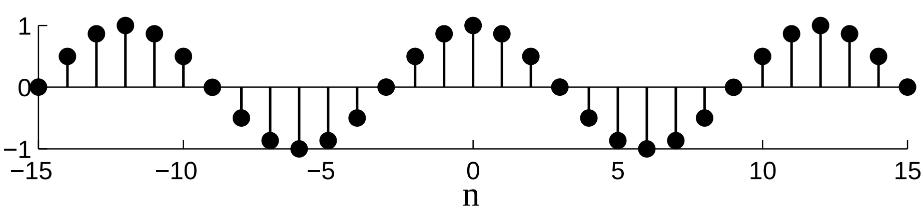
Compared to their continuous-time counterparts (those that take a continuous-valued independent time variable t), discrete-time sinusoidal signals have two unique characteristics. It is possible for them to **alias**, and they are not always **periodic**.

Aliasing of Discrete-time Sinusoids
One might think that if two different discrete-time sinuoids have different frequencies, that they would be different signals. Such is the case with continuous-time sinusoids, but not always for the discrete-time version. Consider two discrete-time sinuoids $x_1[n]$ and $x_2[n]$ with different frequencies, ω and $\omega+2\pi$: $x_1[n]=e^{j(\omega n+\phi)}$ $x_2[n]=e^{j((\omega+2\pi)n +\phi}$ We can then simplify the expression of $x_2[n]$, using the fact that $e^{j2\pi}=1$ to arrive at this surprising conclusion: $x_2[n]=e^{j((\omega+2\pi)n +\phi}=e^{j(\omega n+2\pi n +\phi)}=e^{j(\omega n +\phi)}e^{j2\pi n}=e^{j(\omega n +\phi)}(1)^n=e^{j(\omega n +\phi)}=x_1[n]$ So $x_1[n]$ and $x_2[n]$ had different frequencies, yet they are identical! You can see this plotted out with $\omega=\frac{\pi}{6}$ below:

Here $x_1[n]$ and $x_2[n]$ have different frequencies, yet are identical.



$x_1[n]=\cos\left(\frac{\pi}{6}n\right)$.

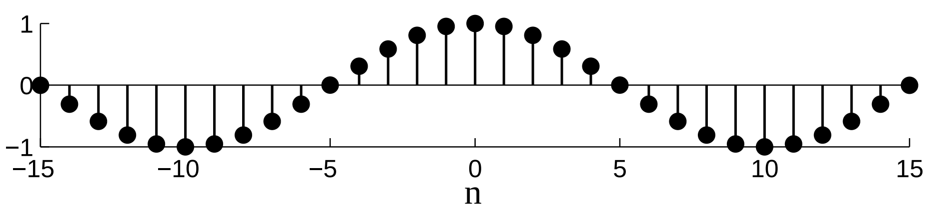


$$x_2[n]=\cos\left(\frac{13\pi}{6}n\right) = \cos\left(\left(\frac{\pi}{6}+2\pi\right)n\right).$$

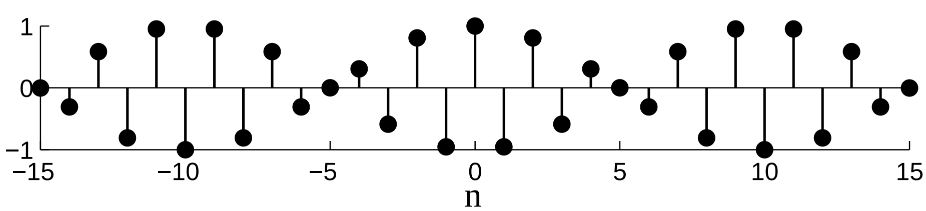
This phenomenon is called **aliasing**. It happens when frequencies are offset by any integer multiple of 2π (you can use $\omega+2\pi m$ in the example above and see for yourself).

So only frequencies along a continuous interval of length 2π on the real number are distinct from each other. For this reason, when we deal with discrete-time frequencies we consider only those along the interval $0\leq\omega<2\pi$ or $-\pi<\omega\leq\pi$, as any other frequency aliases back to an identical signal with a frequency in that range. Within these ranges, frequencies close to 0 (or 2π, depending on the range used) are low frequencies--their sinusoids do not oscillate very quickly--and frequencies close to π (or $-\pi$, depending on the range) are high frequencies:

(a) Low frequencies are those ω close to 0 or 2π rad. (b) High frequencies are those ω close to π or $-\pi$ rad.

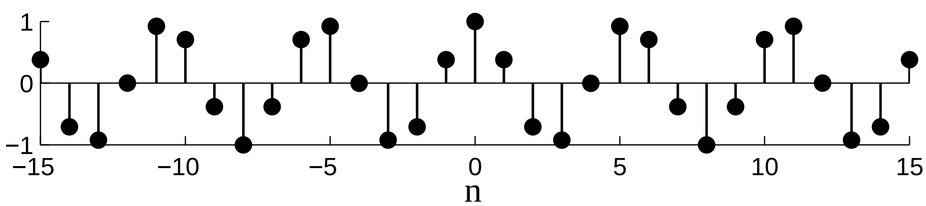


$\cos\left(\frac{\pi}{10}n\right)$.



$\cos\left(\frac{9 \pi}{10}n\right)$.

Periodicity of Discrete-time Sinusoids
Recall that a signal $x[n]$ is defined to be periodic if there exists some integer N for which $x[n+N]=x[n]~,~\forall n$. Suppose we have a complex sinusoid with a frequency $\omega=2\pi\frac{k}{N}$, where k and N are integers. This just means that ω is a fraction of 2π. It turns out that this signal is periodic, with period N: $\begin{align*} x[n]&=e^{j(2\pi\frac{k}{N} n+ \phi)}\\ x[n+N]&=e^{j(2\pi\frac{k}{N} (n+N) + \phi)} \\ &=e^{j(2\pi\frac{k}{N} n + 2\pi\frac{k}{N} N + \phi)}\\ &=e^{j(2\pi\frac{k}{N} n + \phi)} e^{j(2\pi\frac{k}{N} N)}\\ &=e^{j(2\pi\frac{k}{N} n + \phi)}e^{j(2\pi\frac{k}{N}N)}\\ &=e^{j(2\pi\frac{k}{N} n + \phi)}(e^{j(2\pi k)})\\ &=x[n] \end{align*}$ Here is a plot of a sinusoid with frequency $2\pi\frac{3}{16}$. You will note that it has a period of $N=16$:

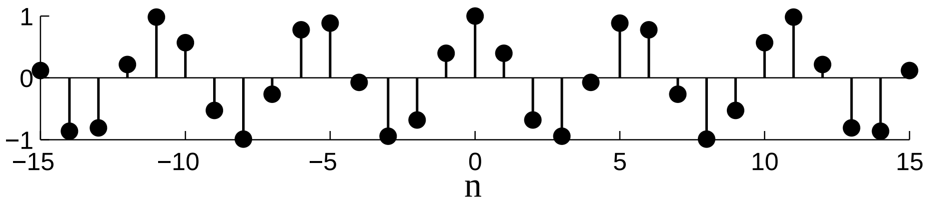


$x_1[n] = \cos(\frac{2\pi 3}{16} n)$ is periodic, with $N=16$.

Now, these fractions of 2π are special values of ω we will call **harmonic frequencies**, for sinusoids with such frequencies are periodic.

In contrast, consider sinusoids whose frequencies are *not* fractions of π: $\begin{align*} x[n]&=e^{j(\omega n+ \phi)}\\ x[n+N]&=e^{j(\omega (n+N) + \phi)} \\ &=e^{j(\omega n + \omega N + \phi)}\\ &=e^{j(\omega n + \phi)} e^{j(\omega N)}\\ &=e^{j(\omega n + \phi)}e^{j(\omega N)}\\ &\neq x[n],\textrm{unless } \omega N=2\pi k \rightarrow \omega=2\pi\frac{k}{N} \end{align*}$ So we see that discrete-time

sinusoids are periodic if, and only if, their frequencies are fractions of 2π. Consider the example of a non-periodic sinusoid below. It definitely oscillates, and at first it appears to be periodic, but look carefully and you will see that it is not, unlike the one from the figure above.



$x_2[n] = \cos(1.16 \, n)$. The frequency of 1.16 is not a fraction of 2π, so this sinusoidal signal is not periodic.

Take note of sinusoids whose frequencies are of the form $\omega=2\pi\frac{k}{N}$, for they will play a starring role in the Fourier analysis of periodic and finite-length discrete-time signals.

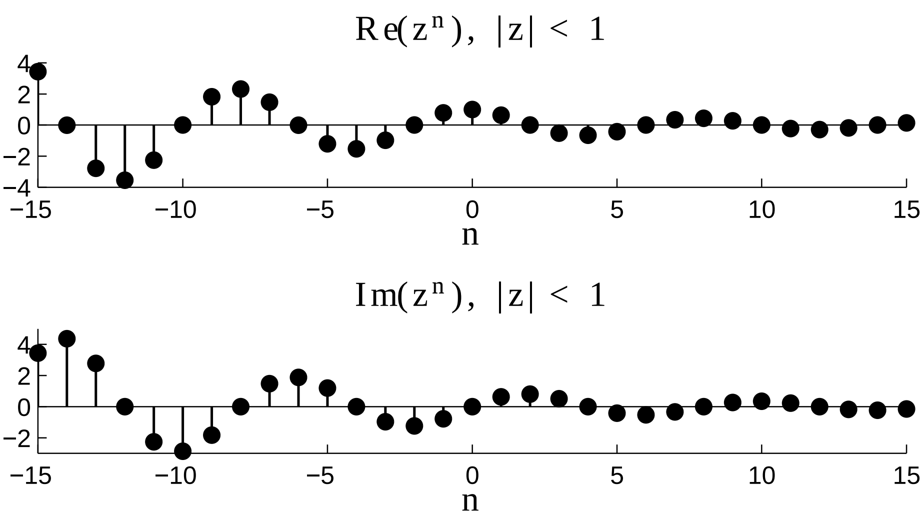
Complex Exponentials

From Complex Sinusoids to Complex Exponentials
Recall the form of a discrete-time complex sinusoid: $x[n]=e^{j(\omega n + \phi)$. As we have already seen, that signal itself is complex-valued, i.e., it has both a real and an imaginary part. But look closely at just the exponent, and you will see that the exponent itself is purely imaginary.

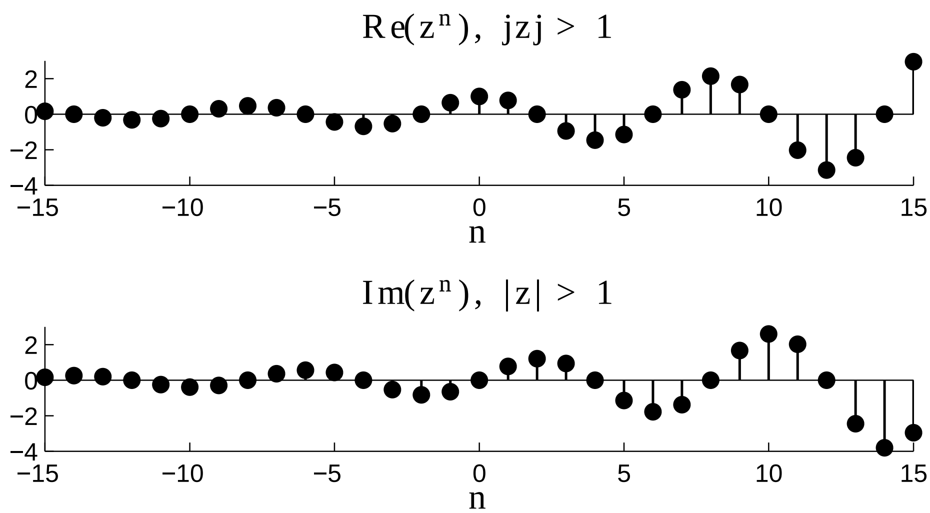
Suppose we let the exponent be complex-valued, say of the form $a+jb$. Then we have $e^{(a+jb)n}=e^{an}e^{jbn}=(e^a)^n e^{jbn}$. So the result is a complex sinusoid multipled by a real exponential signal (whose base is e^a).

Complex Exponentials, Defined
We do not typically represent complex exponentials in the way derived above, but rather express them in the form $x[n]=z^n$, where z is a complex number. Being a complex number, it lies on the complex plane with a magnitude of $|z|$ and an angle of $\angle z$ we define as ω. So then, if we would like to express $x[n]=z^n$ as a combination of a real exponential and a complex sinusoid, as above, we have: $x[n]=z^n=|z|^n e^{j\omega n}$. Below are some plots of complex exponentials for different values of z.

The real and imaginary parts of a complex exponential x^n for which $|z|\lt 1$.



The real and imaginary parts of a complex exponential x^n for which $|z|\gt 1$.



So when the magnitude $|z|$ is greater than 1, we have a signal that oscillates and exponentially grows with time, and if the magnitude is less than 1, it decays over time. And, you guessed it, if the magnitude is exactly equal to 1, it does not grow or decay, but only oscillates. In fact, if the magnitude is 1, the complex exponential is, by definition, simply a complex sinusoid: $|1|^n e^{j\omega n}=e^{j\omega n}$. Therefore you can see that complex sinusoids are a subset of the more general complex exponential signals.